

Detection of Novel Social Bots by Ensembles of Specialized Classifiers

Mohsen Sayyadiharikandeh,^{1*} Onur Varol,² Kai-Cheng Yang,¹ Alessandro Flammini,¹
Filippo Menczer¹

¹Observatory on Social Media, Indiana University, Bloomington, IN, USA

²Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey

ABSTRACT

Malicious actors create inauthentic social media accounts controlled in part by algorithms, known as social bots, to disseminate misinformation and agitate online discussion. While researchers have developed sophisticated methods to detect abuse, novel bots with diverse behaviors evade detection. We show that different types of bots are characterized by different behavioral features. As a result, supervised learning techniques suffer severe performance deterioration when attempting to detect behaviors not observed in the training data. Moreover, tuning these models to recognize novel bots requires retraining with a significant amount of new annotations, which are expensive to obtain. To address these issues, we propose a new supervised learning method that trains classifiers specialized for each class of bots and combines their decisions through the maximum rule. The ensemble of specialized classifiers (ESC) can better generalize, leading to an average improvement of 56% in F1 score for unseen accounts across datasets. Furthermore, novel bot behaviors are learned with fewer labeled examples during retraining. We deployed ESC in the newest version of Botometer, a popular tool to detect social bots in the wild, with a cross-validation AUC of 0.99.

KEYWORDS

Social media, social bots, machine learning, cross-domain, recall

1 INTRODUCTION

Social media accounts partially controlled by algorithms, known as social bots, have been extensively studied [21, 41]. The automated nature of bots makes it easy to achieve scalability when spreading misinformation [36, 37], amplifying popularity [9, 34, 45], or polarizing online discussion [39]. Bot activity has been reported in different domains, including politics [5, 20, 39], health [2, 3, 6, 17], and business [12, 13]. Due to the wide adoption of social media, every aspect of people’s life from news consumption to elections is vulnerable to potential manipulation by bots.

The public is beginning to recognize the existence and role of social bots: according to a recent Pew survey [40], two thirds of Americans have knowledge of bots and over 80% believe bots have a negative impact. Actions have been taken to restrict the potential damage caused by deceptive bots, such as those that pose as humans. For example, California passed a “Bot Disclosure” law in July 2019, requiring bots to reveal themselves in certain cases ([leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB1001](http://leginfo.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB1001)). However, there is no guarantee that such legislative solutions will be

effective against malicious bots, or even that they will survive constitutional challenges. In an arms race between abusive behaviors and countermeasures, novel social bots emerge everyday and evade purge from the platforms [15, 21]. Therefore, the availability of tools to identify social bots is still important for protecting the authenticity and health of the information ecosystem.

Many social bot detection methods based on machine learning have been proposed in the past several years (see Related Work). Here we focus on supervised learning methods, particularly Botometer [43, 48], a widely adopted tool designed to evaluate Twitter accounts. Supervised methods are only as good as their training data. Bots with unseen characteristics are easily missed, as demonstrated by a drastic drop in recall when classifiers are faced with cross-domain accounts [18]. One common approach to address the lack of generalization is to retrain models with new labeled datasets [48]. Unfortunately, high-quality datasets of annotated social media accounts are expensive to acquire.

In this paper, we aim to improve the cross domain performance of Botometer in the wild and better tune the method to the adversarial bot detection problem. Using bot and human accounts in different datasets, we show that bot accounts exhibit greater heterogeneity in their discriminative behavioral features compared to human accounts. Motivated by this observation, we propose a novel method to construct a bot detection system capable of better generalization by training multiple classifiers specialized for different types of bots. Once these domain-specific classifiers are trained, unseen accounts are evaluated by combining their assessments. We evaluate *cross-domain* performance by testing on datasets that are not used for training, as opposed to *in-domain* evaluation through cross-validation. Without loss of in-domain accuracy, the proposed approach effectively increases the recall of cross-domain bots. It can also learn more efficiently from examples in new domains.

Given these results, the proposed method is deployed in the newest version (v4) of Botometer, a widely adopted tool to detect social bots in the wild that is publicly available from the Observatory on Social Media at Indiana University.

2 THE CHALLENGE OF GENERALIZATION

2.1 Datasets

We considered various labeled datasets available through the Bot Repository (botometer.iuni.iu.edu/bot-repository). Most of the datasets are annotated by humans, while others are created using automated techniques based on account behavior, filters on metadata, or more sophisticated procedures to achieve high precision. For example, *astroturf* is a new dataset that includes hyper-active political bots participating in follow trains and/or systematically deleting

*Corresponding Author. Email: msayyadi@indiana.edu.

Table 1: Annotated datasets.

Dataset	Annotation method	Ref.	Bots	Humans
caverlee	Honeypot + verified	26	15,483	14,833
varol-icwsm	Human annotation	43	733	1,495
cresci-17	Various methods	11	7,049	2,764
pronbots	Spam bots	48	17,882	0
celebrity	Celebrity accounts	48	0	5,918
vendor-purchased	Fake followers	48	1,087	0
botometer-feedback	Human annotation	48	139	380
political-bots	Human annotation	48	62	0
gilani-17	Human annotation	22	1,090	1,413
cresci-rtbust	Human annotation	28	353	340
cresci-stock	Sign of coordination	12	7,102	6,174
botwiki	Human annotation	49	698	0
astroturf	Human annotation		505	0
midterm-2018	Human annotation	49	0	7459
kaiser-1	Politicians + new bots	35	875	499
kaiser-2	German politicians + German bots	35	27	508
kaiser-3	German politicians + new bots	35	875	433
combined-test	gilani-17	+	9,432	8,862
	cresci-rtbust	+		
	cresci-stock	+		
	kaiser-1 + kaiser-2			

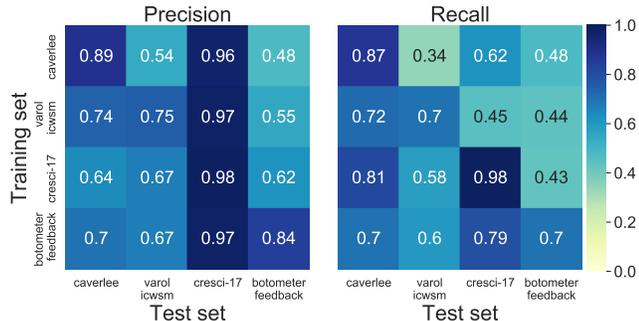


Figure 1: Precision (left) and recall (right) of Random Forests trained on one dataset (row) and tested on another (column).

content. Detailed dataset descriptions are outside the scope of the present paper, but summary statistics and references can be found in Table 1. In addition to the datasets in the Bot Repository, we also collected accounts provided in a recent study by Rauchfleisch and Kaiser [35]. These datasets made an assumption that all accounts belonging to American and German politicians are human accounts. They complement this dataset with manually annotated German language bots and accounts listed in the botwiki dataset.

For training models, we extract over 1,200 features in six categories: metadata from the accounts and friends, retweet/mention networks, temporal features, content information, as well as sentiment. These features are shown to be effective in identifying social bots and are described in detail in the literature [42, 43, 48].

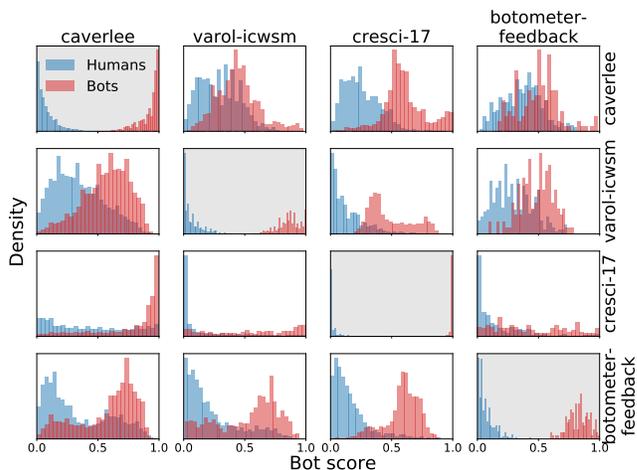


Figure 2: Bot score distributions for human (blue) and bot (red) accounts in each experiment. Models used in these experiments are trained on the datasets labeled along the rows and tested on the datasets listed in the columns.

2.2 Cross-domain performance comparison

Supervised bot detection methods achieve high accuracy based on in-domain cross-validation [4]. To measure how recall deteriorates in cross-domain evaluation, we perform an experiment using four datasets selected from Table 1: we train a model on one dataset and test it on another. We use Random Forest classifiers with 100 decision trees (similar to the baseline model described in § 3.1). The fraction of trees outputting a positive label is calibrated using Platt’s scaling [31] and binarized with a threshold of 0.5. We use 5-fold cross-validation for in-domain classification; for consistency we split training and test samples in cross-domain cases as well, reporting average precision and recall.

The results of our experiment are shown in Fig. 1. Diagonal (off-diagonal) cells represent in-domain (cross-domain) performance. Both precision and recall tend to be higher for in-domain cases, demonstrating the limited generalization of supervised models across domains. The one exception is the high precision when testing on the cresci-17 domain, irrespective of the training datasets. This is due to the fact that cresci-17 includes spambots, which are represented in all datasets. By comparing the two panels, we see that recall of bots is more impaired in cross-domain tests, in line with previous findings [11, 18]. The method proposed here improves cross-domain bot recall.

To interpret the cross-domain classification results, we plot the distributions of bot scores in Fig. 2. A bot score is the output of a Random Forest classifier and corresponds to the proportion of decision trees in the ensemble that categorize the account as a bot. In the diagonal plots (in-domain tests), the density plots are left-skewed for humans and right-skewed for bots, representing a good separation and yielding high precision, recall, and F1. For most of the cross-domain experiments, the score distributions are still left-skewed for humans, but not right-skewed for bots. This suggests that bot accounts tend to have lower cross-domain scores,

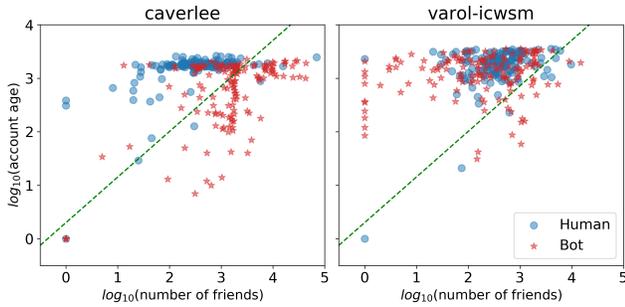


Figure 3: Separation of bots and humans based on two features in different datasets. Both plots show the logistic-regression decision boundary obtained from *caverlee*.

resulting in lower recall. Human accounts, on the other hand, exhibit consistent characteristics across datasets. This observation suggests a way to improve generalization.

2.3 Predictability of different bot classes

There are different kinds of bots. Consider the three different bot classes in the *cresci-17* dataset: traditional spambots, social spambots, and fake followers. We trained decision trees to discriminate each of these classes of bots from the others. Table 2 shows that different features are most informative for each class: traditional spambots generate a lot of content promoting products and can be detected by the frequent use of adjectives; social spambots tend to attack or support political candidates, therefore sentiment is an informative signal; finally, fake followers tend to have aggressive following patterns, flagged by the friend/follower ratio.

Given such heterogeneity of bot behaviors, we conjecture that the drop in cross-domain recall can be attributed to the distinct discriminating features of accounts in different datasets. To explore this conjecture, let us use the Gini impurity score of a Random Forest classifier to find the two most informative features for the *caverlee* dataset, then train a logistic regression model on those two features. Fig. 3 visualizes bot and human accounts in *caverlee* and *varol-icwsm* on the plane defined by the two features. For the in-domain case (left), the linear classifier is sufficient to separate human and bot accounts. But in the cross-domain case (right), the same linear model fails, explaining the drop in recall: different features and distinct decision rules are needed to detect different classes of bots.

3 METHODS

3.1 Baseline bot detection models

Before presenting the proposed method, let us select two baselines for evaluation. The first baseline is the current version of Botometer, often considered the state-of-the-art method for bot detection [48]. The model is a Random Forest, a classification model that has proven to be effective in high-dimensional problems with complex decision boundaries. In this approach, we output the fraction of positive votes as a *bot score*. Bots from all datasets are merged into a single bot (positive) class. We refer to this baseline as *Botometer-v3*. We also

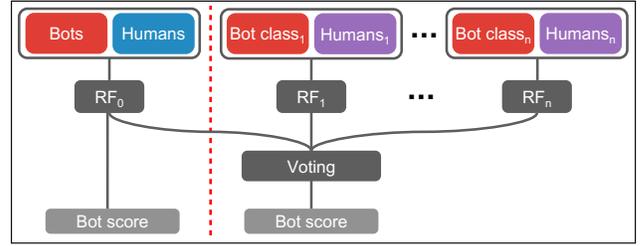


Figure 4: Illustration of the proposed model. The bot score from RF_0 corresponds to the previous version of Botometer, that from the voting module to the new (ESC) version.

consider a variation of the Botometer baseline that does not consider a set of features describing time zone and language metadata, as those are no longer available through the Twitter API. We refer to this variation as *Botometer-v3.1*. The two Botometer baselines use 1,209 and 1,160 features, respectively.

We use *tweetbotornot2* as a second baseline. This model is based on a supervised classifier that considers over a hundred features in three main categories: user-level attributes, tweeting statistics, and text-based patterns. The motivation behind this choice of baseline is that *tweetbotornot2* has been developed independently, is widely used, and is easily accessible by the general public via an R library ([tweetbotornot2.mikewk.com](https://github.com/mikewk/tweetbotornot2)).

3.2 Proposed method

The proposed approach is inspired by the two empirical findings discussed in the previous section. First, inspired by the observation that human accounts are more homogeneous than bots across domains, we train a model on all human and bot examples across datasets and use it to identify likely humans. Second, since different bot classes have different sets of informative features, we propose to build specialized models for distinct bot classes. The specialized human and bot models are aggregated into an ensemble and their outputs are combined through a voting scheme. We call this approach *Ensemble of Specialized Classifiers (ESC)*.

Fig. 4 illustrates the ESC architecture. The human detection subsystem actually corresponds to Botometer (the baseline classifier), and constitutes the left-most component RF_0 of the ensemble shown in the figure. We then build specialized bot classifiers using Random Forest models ($RF_1 \dots RF_n$ in Fig. 4). We use 100 decision tree estimators; all other parameters take the default values. Each specialized classifier RF_i is trained on a balanced set of accounts from bot class BC_i and an equal number $|BC_i|$ of human examples sampled from human accounts across all datasets.

A bot score is calculated by a voting scheme for the classifiers in the ensemble. Among the specialized bot classifiers, the one that outputs the highest bot score s_i is most likely to have recognized a bot of the corresponding class. Therefore we use the maximum rule to aggregate the bot scores. For the human classifier RF_0 , a low bot score s_0 is a strong signal of a human account. Therefore we determine the winning class as $i^* = \arg \max_i \{s'_i\}$ where

$$s'_i = \begin{cases} 1 - s_i & \text{if } i = 0 \\ s_i & \text{else.} \end{cases}$$

Table 2: Most informative features per bot class in *cresci-17*.

Rank	Traditional spambots	Social spambots	Fake followers
1	Std. deviation of adjective frequency	Tweet sentiment arousal entropy	Max. friend-follower ratio
2	Mean follower count	Mean friend count	Std. deviation of tweet inter-event time
3	Tweet content word entropy	Mean adjective frequency	Mean follower count
4	Max. friend-follower ratio	Minimum favorite count	User tweet-retweet ratio
5	Max. number of retweet count	Tweet content word entropy	Mean tweet sentiment happiness

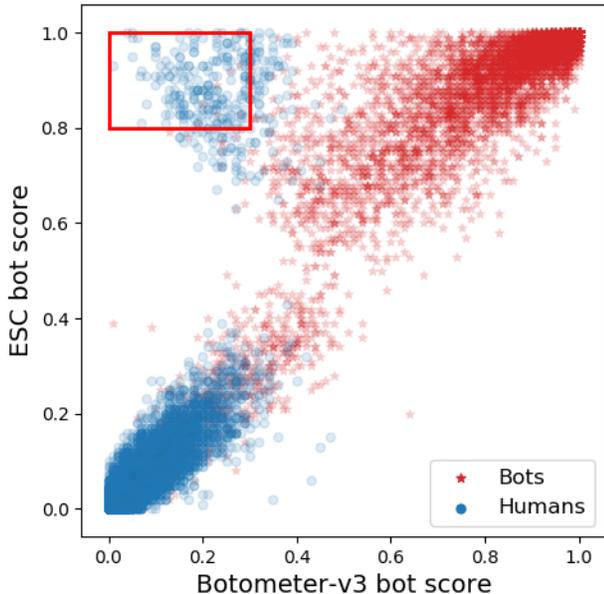


Figure 5: Correlation between bot scores obtained from the two methods.

The ESC bot score is obtained by calibrating the score s_{i^*} using Platt’s scaling [31]. As we see in § 4, the maximum rule has the effect of shifting scores of likely bots toward one and the scores of likely humans toward zero. Along with the bot score, ESC can also produce the bot class label i^* as an explanatory outcome.

4 RESULTS

To train the specialized classifiers, we need coherent bot behaviors. For the experiments in this section, we organized the bot accounts in the training data into separate classes of bots: simple bots, spammers, fake followers, self-declared, political bots, and others. Simple bots are derived from *caverlee*. For spammer bots we use bot accounts in *pron-bots* and a subset of *cresci-17*. Fake followers include subsets of bot accounts in *cresci-17* and *vendor-purchased*. Self-declared bots are derived from *botwiki*. Political bots come from *political-bots* plus a subset of *astroturf*. The rest of the bots captured in other datasets are aggregated into the “others” category.

4.1 In-domain performance

Before discussing cross-domain performance, let us demonstrate that ESC is capable of detecting bots with good accuracy in the

classic (in-domain) scenario. Using 5-fold cross-validation, ESC achieves an Area Under the ROC Curve (AUC) of 0.96, similar to the Botometer classifier (0.97 AUC). The scatter plot in Fig. 5 shows a good agreement between the bot scores obtained with ESC and the *Botometer-v3* baseline (Spearman’s $\rho = 0.87$).

Let us pay special attention to accounts having low Botometer and high ESC scores, in other words those that are likely human according to Botometer, but likely bots according to ESC (region highlighted in Fig. 5). These are the only cases where we observe a clear disagreement between the two methods on ground-truth labels. We focus on 332 accounts in this region that are labeled as human (75% are from *caverlee*), which represent approximately 1% of the examples labeled as human in the training data. One possible interpretation of the disagreement is that these accounts are incorrectly classified by ESC (false positives). Another possibility is that some of these accounts may have changed since they were manually labeled. Indeed, training datasets are subject to change over time as accounts become inactive, suspended, or get compromised by third-party applications. Such changes could lead to errors on ground-truth labels.

Manual inspection of a random sample of 50 of the accounts highlighted in Fig. 5 reveals that the human labels are no longer accurate for most of them — they have been inactive for years, are currently devoted to spam diffusion, and/or are controlled by third-party applications. This suggests that ESC can identify impurities in the training data. While mislabeled accounts impair the performance of machine learning models, we conjecture that the ESC model is still able to recognize them because it is more robust to errors — the incorrect labels only affect a subset of the classifiers.

4.2 Cross-domain performance

We want to demonstrate that the proposed ESC approach generalizes better to cross-domain accounts compared to the current version of Botometer. In this set of experiments, some datasets are held out in the training phase and are then used as cross-domain test cases: *cresci-stock*, *gilani-17*, *cresci-rtbust*, *kaiser-1*, *kaiser-2*, and *kaiser-3*. In addition, *combined-test* combines these datasets while avoiding duplication. We focus on F1 and AUC as the accuracy metrics. We obtain confidence intervals using bootstrapping with five samples of 80% of the accounts in the test set.

We compare the ESC approach with the *Botometer-v3* baseline model [48], as well as the *Botometer-v3.1* baseline to exclude the possibility that improvements are due to the removal of features based on deprecated metadata. The language agnostics version of Botometer (excluding English-based linguistic features) is used on the *kaiser-2* dataset because of its German tweet content.

To illustrate the main enhancement afforded by ESC, let us compare the distributions of bot scores generated by *Botometer-v3* and

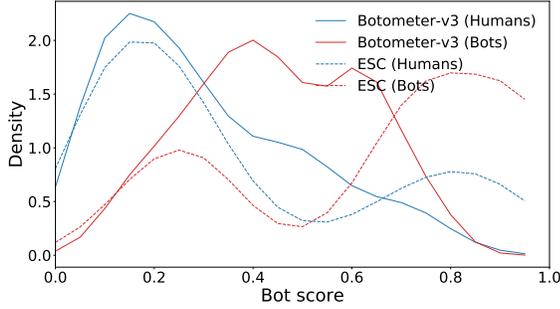


Figure 6: Distributions of bot scores (using KDE) for both methods on the hold-out `cresci-rtbust` dataset.

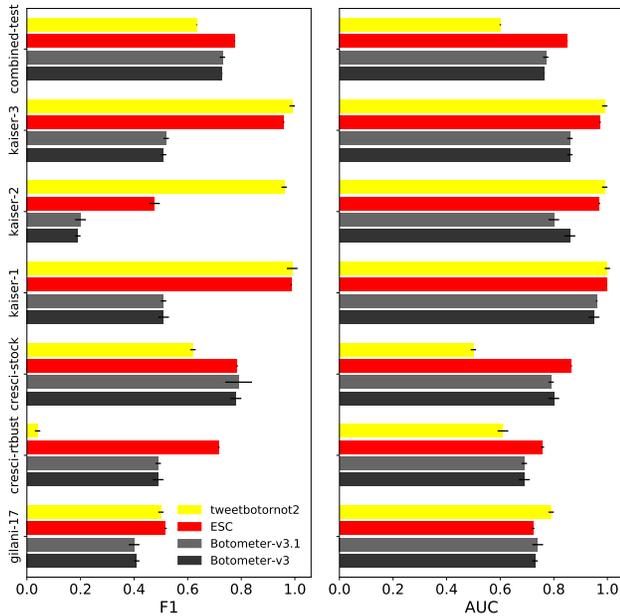


Figure 7: F1 (left) and AUC (right) of ESC and baseline methods on hold-out test datasets. Error bars indicate 95% confidence intervals. Note that `kaiser-1` and `kaiser-3` are not ‘hold-out’ as there is overlap with the training sets (see text).

ESC in a cross-domain experiment. Due to space limitations, we illustrate in Fig. 6 just one case where `cresci-rtbust` is used as test set; other cases are similar. Both methods tend to yield low scores for human accounts, as the same classifier (RF_0) is used. On the other hand, ESC produces significantly higher scores than *Botometer-v3* on bot accounts. This is a result of the maximum rule and leads to higher cross-domain recall, or better generalization.

Fig. 7 shows that ESC outperforms the Botometer baseline in most cases. On average across the six datasets, recall goes from 42% to 84% (an improvement of 100%) while precision increases from 52% to 64%. As a result, F1 increases from 47% to 73% (an improvement of 56%). On the combined-test dataset, recall goes

from 77% to 86%, precision stays at 70%, and F1 goes from 73% to 77% (an improvement of 5%). Comparisons based on AUC scores are similar.

The `kaiser-1` and `kaiser-3` datasets include bots from `botwiki`, which are part of the ESC training data. Therefore these two cannot be considered completely hold-out datasets, but are included nonetheless because they were used to highlight weaknesses of *Botometer-v3* in a recent independent report [35], so they provide us with an opportunity to demonstrate the performance of the latest Botometer model. Even if we exclude `botwiki` from the training data, ESC still outperforms *Botometer-v3*. For example, it yields an F1 score of 0.84 on `kaiser-1` and 0.80 on `kaiser-3`.

ESC yields F1 better than or comparable with *tweetbotornot2* on all datasets except those from `Rauchfleisch` and `Kaiser` [35]. The AUC metric is comparable on those datasets, while *tweetbotornot2* wins on `gilani-17` and ESC wins on the other datasets. On the combined-test dataset, ESC outperforms *tweetbotornot2* on both F1 and AUC.

In interpreting these results, note that `kaiser-3` contains human accounts from `kaiser-2` and bot accounts from `kaiser-1`, so they are not independent. `kaiser-2` includes only 27 bot accounts; the F1 score is sensitive to this class imbalance. ESC is comparable to *tweetbotornot2* on this dataset when using the AUC metric, which is not sensitive to class imbalance. Furthermore, while we do not know how the *tweetbotornot2* baseline was trained, it uses a feature for ‘verified’ profiles and tends to assign a low bot score to them, even automated ones such as `@twitter-support`. 98% and 72% of accounts labeled human in `kaiser-1` and `kaiser-3` respectively are verified. This biases the results in favor of *tweetbotornot2*.

4.3 Model adaptation

Real-world applications of social bot detection always face the challenge of recognizing novel behaviors that are not represented in the training data. Periodic retraining to include newly annotated datasets helps systems adapt to these unseen examples. A common approach is to train a new classifier from scratch including both old and new training data, which may not be efficient. The proposed ESC method alleviates this problem because we can add a new classifier RF_{n+1} to the ensemble to be trained with the new data, without retraining the existing classifiers.

Let analyze how quickly the *Botometer-v3* and ESC models adapt to a new domain. To quantify this, we split the data from a hold-out domain into two random subsets for training and testing. Results are presented using `varol-icwsm` as the hold-out domain for both models. (We reach similar conclusions using other hold-out datasets.) 800 examples from the hold-out dataset are used for training. In each iteration we randomly sample 50 examples and add them to the training set. In the *Botometer-v3* case, we retrain the entire classifier (RF_0), whereas in the ESC case we only train a newly added specialized classifier (RF_{n+1}).

Fig. 8 shows how the F1 score on the test data improves as a function of the number of training examples from the hold-out domain. The *Botometer-v3* baseline adapts more slowly. We can interpret this result by recalling that the size of the hold-out dataset is small compared to the training size (over 67,000 examples in *Botometer-v3*). The decision trees use Gini gain as a feature selection criterion,

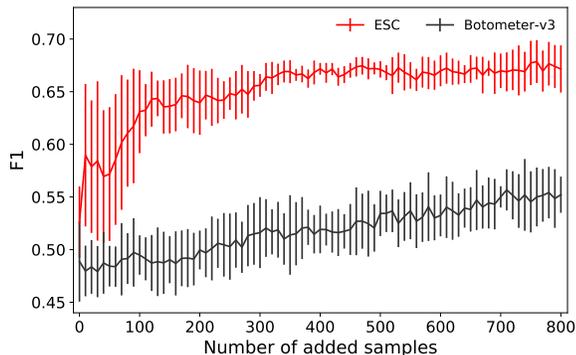


Figure 8: Adaptation of two methods to new examples from the varo1-icwsm dataset, which was held out from the training data.

therefore the number of examples sharing the same informative features affects the selection of those features. As a result, the old bot classes with more examples dominate and the classifier struggles to learn about the new domain. On the other hand, the ESC architecture quickly learns about new bots through the new classifier, which starts from scratch, while what was learned about the old ones is preserved in the existing classifiers. This means that fewer labeled examples are needed to train a new specialized classifier when novel types of bots are observed in the wild.

5 RELATED WORK

Different approaches have been proposed for automatic social bot detection. Crowdsourcing is convenient and one of the first proposals to collect annotated data effectively [47], but annotation has limitations due to scalability and user privacy. Thus automatic methods are of greater interest, especially for social media services that deal with millions of accounts. The structure of the social graph captures valuable connectivity information. Facebook employed an immune system [38] that relied on the assumption that sybil accounts tend to connect mostly to other sybil accounts while having a small number of links to legitimate accounts.

Supervised machine learning approaches, such as the one proposed in this paper, extract various features from an account’s profile, social network, and content [16, 23, 29, 41, 43, 48, 49]. These methods rely on annotated datasets for learning the difference between bot and human accounts. However, since bots change and evolve continuously to evade detection, supervised learning algorithms need to adapt to new classes of bots [21, 44, 48].

Some unsupervised learning methods have been proposed in the literature [11, 27]. They can be less vulnerable to performance decay across domains. They are especially suitable for finding coordination among bots [1, 29, 32]. Since accounts in a coordinated botnet may not appear suspicious when considered individually, supervised methods would miss them [11, 24]. Identifying botnets requires analysis of the activity of multiple accounts to reveal their coordination. Depending on the type of bots, similarity can be

detected through tweet content [8, 25], temporal features in the timelines [7, 10], or retweeting behavior [28, 46].

A recent research direction is to address the limits of current bot detection frameworks in an adversarial setting. Cresci et al. [14] predict that future techniques will be able to anticipate the ever-evolving spambots rather than taking countermeasures only after seeing them. The performance decay of current detection systems in the wild was reported by Cresci et al. [11], who showed that Twitter, human annotators, and state-of-the-art bot detection systems failed at discriminating between some new social spambots and genuine accounts. In agreement with the present findings, Echeverria and Zhou [19] suggest that detecting different types of bots requires different types of features. Echeverria et al. [18] proposed the leave-one-botnet-out methodology, to highlight how detection methods do not generalize well to unseen bot classes due to bias in the training data. Even a classifier trained on 1.5 million accounts and 22 classes of bots is incapable of identifying new classes of bots. Here we follow the leave-one-class-out methodology to evaluate the generalization power of the proposed approach.

Some papers have characterized bot classes [30]. Lee et al. [26] define traditional spammers, mention spammers, friend infiltrators, and social spammers in their dataset. Cresci et al. [11] highlight that it is hard for human annotators to assign one label to one account to describe its bot class. They also report that some bot classes, like social spambots, are hard to distinguish from human accounts. Despite the existence of different types of bots, no systems have been presented in the literature to automatically identify the type of a bot as the method proposed here.

Rauchfleisch and Kaiser [35] have criticised Botometer (more specifically *Botometer-v3*) for its high false positive and false negative rates. As we have discussed in this paper, we share these concerns and acknowledge that like any supervised learning model, Botometer makes mistakes. Indeed, the new version introduced in the paper is motivated by this issue and partly addresses it by improving cross-domain recall (false negatives). At the same time, Rauchfleisch and Kaiser [35] may overestimate the false positive rate by assuming that no politician account uses automation. We believe this assumption is not realistic, considering these accounts are often managed by media teams and use scheduling tools for content creation. In fact, Botometer currently does not use the “verified” status as a feature because it could lead to false negatives [45]. Another source of bias is that Rauchfleisch and Kaiser [35] overlook accounts that are no longer available due to suspension, possibly leading to an underestimation of both precision and recall.

6 BOTOMETER-V4 DEPLOYMENT

In light of these results, we are deploying ESC in the newest version of Botometer (v4), a tool to detect social bots in the wild that is available through the Observatory on Social Media (OSoMe) at Indiana University. Botometer can be accessed both through an interactive website (botometer.org) and programmatically through a public API (rapidapi.com/OSoMe/api/botometer-pro). Fig. 9 illustrates the system architecture.

The deployed system is implemented in Python with the MKL library for efficiency. Random Forest classifiers are implemented

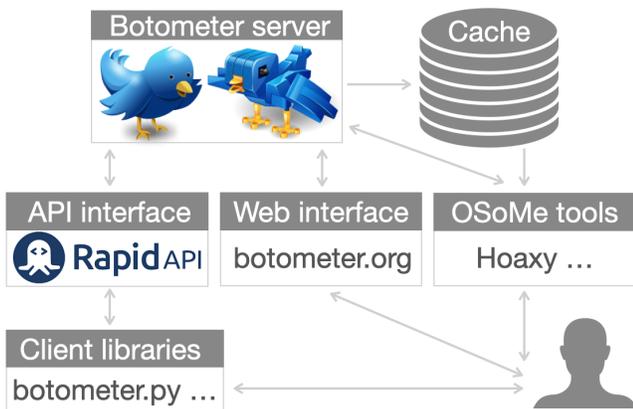


Figure 9: The architecture of the Botometer ecosystem.

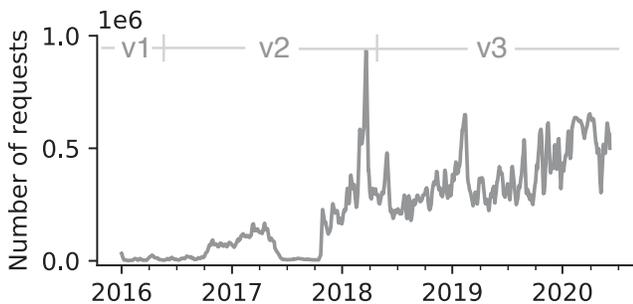


Figure 10: Daily requests of the Botometer API. Annotations indicate the versions of the models deployed in different time periods.

using the *scikit-learn* package [33]. Since ESC provides a good correspondence between scores and binary labels of human and bot accounts, no calibration is applied. We performed load tests submitting queries from 20 simultaneous jobs running on a supercomputer, yielding an average response time below 100ms per query. This should provide ample scalability, considering the popularity of the tool among researchers and practitioners — at the time of this writing, it fields over half a million queries daily (Fig. 10).

To train the specialized classifiers of the deployed model on homogeneous bot types, we rearranged the datasets in Table 1. The classes are similar to those described in § 4, with several modifications: (i) we removed *caverlee* based on the analysis in Fig. 5; (ii) we used the full *astroturf* dataset; (iii) we added a financial bot class based on *cresci-stock*; (iv) we extended the class of other bots using *cresci-rtbust*, *gilani-17*, the new bots in *kaiser-1*, and the German bots in *kaiser-2*; and (v) we added human accounts from the combined-test dataset and *midterm-2018*. The final model yields an AUC of 0.99.

The new front-end of the Botometer website and the Botometer API report the scores obtained from the six specialized classifiers (fake followers, spammers, self-declared, astroturf, financial, others). In this way, the tool offers greater transparency about the decision

process by allowing inspection of the outputs of different ensemble components and by providing interpretable class labels in output.

7 CONCLUSION

The dynamic nature of social media creates challenges for machine learning systems making inferences and predictions on online data. On the one hand, platforms can change features, require models to be retrained. Further difficulties arise as accounts used for training change behavior, become inactive, compromised, or are removed from the platform, invalidating ground-truth data. On the other hand, account behaviors can change and evolve. As is typical in adversarial settings, automated accounts become more sophisticated to evade detection. The emergence of more advanced bot capabilities brings additional challenges for existing systems that struggle to generalize to novel behaviors.

Despite impressive results when training and test sets are from the same domain — even using cross-validation — supervised models will miss new classes of bots, leading to low recall. We demonstrate in this paper that the performance deterioration observed for out-of-domain accounts is due to heterogeneous bot behaviors that require different informative subsets of features. Inspired by this, we presented a novel approach for the automatic identification of novel social bots through an ensemble of specialized classifiers trained on different bot classes. We demonstrated empirically that our proposed approach generalizes better than a monolithic classifier and is more robust to mislabeled training examples. However, our experiments show that cross-domain performance is highly sensitive to the datasets used in training and testing supervised models; it is easy to cherry-pick examples that make any given method look bad.

The proposed architecture is highly modular as each specialized classifier works independently, so one can substitute any part with different models as needed. We can also include additional specialized classifiers when new annotated datasets become available. We showed that this approach allows the system to learn about new domains in an efficient way, in the sense that fewer annotated examples are necessary.

In future work, we would like to investigate methods to recognize the appearance of a new type of bots that warrants the addition of a new classifier to the ensemble. We could also design an unsupervised method to cluster similar accounts across datasets automatically, and assign homogeneous accounts to each of the specialized classifiers. Finally, one could design active learning query strategies to make the retraining process even more efficient than with random selection. This would be useful when reliable user feedback is available to be used as an oracle.

8 ACKNOWLEDGMENTS

We are grateful to Clayton A. Davis and Emilio Ferrara who contributed to early versions of Botometer, and Chris Torres-Lugo for helping with the *astroturf* dataset. This work was supported in part by DARPA (grant W911NF-17-C-0094), Knight Foundation, and Craig Newmark Philanthropies.

REFERENCES

- [1] Faraz Ahmed and Muhammad Abulaish. 2013. A generic statistical approach for spam detection in Online Social Networks. *Computer Communications* 36, 10-11 (2013), 1120–1129.
- [2] Jon-Patrick Allem and Emilio Ferrara. 2018. Could social bots pose a threat to public health? *American Journal of Public Health* 108, 8 (2018), 1005.
- [3] Jon-Patrick Allem, Emilio Ferrara, Sree Priyanka Uppu, Tess Boley Cruz, and Jennifer B Unger. 2017. E-cigarette surveillance with social media data: social bots, emerging topics, and trends. *JMIR public health and surveillance* 3, 4 (2017), e98.
- [4] Eiman Alothali, Nazar Zaki, Elfadil A Mohamed, and Hany Alashwal. 2018. Detecting social bots on Twitter: a literature review. In *International Conference on Innovations in Information Technology*. IEEE, 175–180.
- [5] Alessandro Bessi and Emilio Ferrara. 2016. Social bots distort the 2016 US Presidential election online discussion. *First Monday* 21, 11 (2016).
- [6] David A Broniatowski, Amelia M Jamison, SiHua Qi, Lulwah AlKulaib, Tao Chen, Adrian Benton, Sandra C Quinn, and Mark Dredze. 2018. Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate. *American Journal of Public Health* 108, 10 (2018), 1378–1384.
- [7] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. 2016. DeBot: Twitter Bot Detection via Warped Correlation. In *Proc. Intl. Conf. on Data Mining*. 817–822.
- [8] Zhouhan Chen and Devika Subramanian. 2018. An Unsupervised Approach to Detect Spam Campaigns that Use Botnets on Twitter. *arXiv preprint arXiv:1804.05232* (2018).
- [9] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2015. Fame for sale: efficient detection of fake Twitter followers. *Decision Support Systems* 80 (2015), 56–71.
- [10] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2016. DNA-inspired online behavioral modeling and its application to spambot detection. *IEEE Intelligent Systems* 31, 5 (2016), 58–64.
- [11] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2017. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proc. Intl. Conf. of the Web Companion*. 963–972.
- [12] Stefano Cresci, Fabrizio Lillo, Daniele Regoli, Serena Tardelli, and Maurizio Tesconi. 2018. \$ FAKE: Evidence of Spam and Bot Activity in Stock Microblogs on Twitter. In *Proc. 12th International AAAI Conference on Web and Social Media*.
- [13] Stefano Cresci, Fabrizio Lillo, Daniele Regoli, Serena Tardelli, and Maurizio Tesconi. 2019. Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter. *ACM Transactions on the Web (TWEB)* 13, 2 (2019), 11.
- [14] Stefano Cresci, Marinella Petrocchi, Angelo Spognardi, and Stefano Tognazzi. 2018. From Reaction to Proaction: Unexplored Ways to the Detection of Evolving Spambots. In *Companion of The Web Conference*. 1469–1470.
- [15] Stefano Cresci, Marinella Petrocchi, Angelo Spognardi, and Stefano Tognazzi. 2019. Better safe than sorry: an adversarial approach to improve social bot detection. In *Proceedings of the 10th ACM Conference on Web Science*. 47–56.
- [16] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2016. BotOrNot: A system to evaluate social bots. In *In Proc. 25th Intl. Conf. Companion on World Wide Web*. 273–274.
- [17] Ashok Deb, Anuja Majmundar, Sungyong Seo, Akira Matsui, Rajat Tandon, Shen Yan, Jon-Patrick Allem, and Emilio Ferrara. 2018. Social Bots for Online Public Health Interventions. In *Proc. of the Intl. Conf. on Advances in Social Networks Analysis and Mining*.
- [18] Juan Echeverría, Emiliano De Cristoforo, Nicolas Kourtellis, Ilias Leontiadis, Gianluca Stringhini, Shi Zhou, et al. 2018. LOBO—evaluation of generalization deficiencies in Twitter bot classifiers. In *Proc. of the Annual Computer Security Applications Conf*. ACM, 137–146.
- [19] Juan Echeverria and Shi Zhou. 2017. Discovery of the Twitter Bursty Botnet. *arXiv preprint arXiv:1709.06740* (2017).
- [20] Emilio Ferrara. 2017. Disinformation and Social Bot Operations in the Run Up to the 2017 French Presidential Election. *First Monday* 22, 8 (2017).
- [21] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Commun. ACM* 59, 7 (2016), 96–104.
- [22] Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, Liang Wang, and Jon Crowcroft. 2017. Of bots and humans (on Twitter). In *Proc. of the Intl. Conf. on Advances in Social Networks Analysis and Mining*. ACM, 349–354.
- [23] Zafar Gilani, Ekaterina Kochmar, and Jon Crowcroft. 2017. Classification of Twitter accounts into automated agents and human users. In *Proc. of the Intl. Conf. on Advances in Social Networks Analysis and Mining*. ACM, 489–496.
- [24] Christian Grimme, Dennis Assenmacher, and Lena Adam. 2018. Changing Perspectives: Is It Sufficient to Detect Social Bots?. In *Proc. International Conference on Social Computing and Social Media*. 445–461.
- [25] Sneha Kudugunta and Emilio Ferrara. 2018. Deep Neural Networks for Bot Detection. *Information Sciences* 467, October (2018), 312–322.
- [26] Kyumin Lee, Brian David Eoff, and James Caverlee. 2011. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. In *Proc. AAAI Intl. Conf. on Web and Social Media (ICWSM)*.
- [27] Shenghua Liu, Bryan Hooi, and Christos Faloutsos. 2017. HoloScope: Topology-and-Spike Aware Fraud Detection. *CoRR abs/1705.02505* (2017). arXiv:1705.02505 <http://arxiv.org/abs/1705.02505>
- [28] Michele Mazza, Stefano Cresci, Marco Avvenuti, Walter Quattrociocchi, and Maurizio Tesconi. 2019. RTbust: Exploiting Temporal Patterns for Botnet Detection on Twitter. *arXiv preprint arXiv:1902.04506* (2019).
- [29] Zachary Miller, Brian Dickinson, William Deitrick, Wei Hu, and Alex Hai Wang. 2014. Twitter spammer detection using data stream clustering. *Information Sciences* 260 (2014), 64–73.
- [30] Silvia Mitter, Claudia Wagner, and Markus Strohmaier. 2014. A categorization scheme for socialbot attacks in online social networks. *CoRR abs/1402.6288* (2014). arXiv:1402.6288 <http://arxiv.org/abs/1402.6288>
- [31] Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting Good Probabilities with Supervised Learning. In *Proc. 22nd International Conference on Machine Learning (ICML)* (Bonn, Germany). 625–632.
- [32] Diogo Pacheco, Pik-Mai Hui, Christopher Torres-Lugo, Bao Tran Truong, Alessandro Flammini, and Filippo Menczer. 2020. Uncovering Coordinated Networks on Social Media. *preprint arXiv:2001.05658* (2020). To appear in Proc. ICWSM 2021.
- [33] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.
- [34] Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. 2011. Truthy: mapping the spread of astroturf in microblog streams. In *Proc. 20th Intl. Conf. Companion on World Wide Web*. 249–252.
- [35] Adrian Rauchfleisch and Jonas Kaiser. 2020. The False Positive Problem of Automatic Bot Detection in Social Science Research. *SSRN Electronic Journal* (01 2020). <https://doi.org/10.2139/ssrn.3565233>
- [36] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2018. The spread of low-credibility content by social bots. *Nature communications* 9, 1 (2018), 4787.
- [37] Chengcheng Shao, Pik-Mai Hui, Lei Wang, Xinwen Jiang, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. 2018. Anatomy of an online misinformation network. *PLoS ONE* 13, 4 (2018), e0196087.
- [38] Tao Stein, Erdong Chen, and Karan Mangla. 2011. Facebook Immune System. In *Proc. 4th Workshop on Social Network Systems (SNS)* (Salzburg, Austria). Article 8, 8 pages. <https://doi.org/10.1145/1989656.1989664>
- [39] Massimo Stella, Emilio Ferrara, and Manlio De Domenico. 2018. Bots increase exposure to negative and inflammatory content in online social systems. *Proceedings of the National Academy of Sciences* 115, 49 (2018), 12435–12440.
- [40] Galen Stoking and Nami Sumida. 2018. Social Media Bots Draw Public’s Attention and Concern. *Pew Research Center* (15 Oct 2018). <https://www.journalism.org/2018/10/15/social-media-bots-draw-publics-attention-and-concern/>
- [41] VS Subrahmanian, Amos Azaria, Skylar Durst, Vadim Kagan, Aram Galstyan, Kristina Lerman, Linhong Zhu, Emilio Ferrara, Alessandro Flammini, Filippo Menczer, et al. 2016. The DARPA Twitter bot challenge. *Computer* 49, 6 (2016), 38–46.
- [42] Onur Varol, Clayton A Davis, Filippo Menczer, and Alessandro Flammini. 2018. Feature Engineering for Social Bot Detection. *Feature Engineering for Machine Learning and Data Analytics* (2018), 311–334.
- [43] Onur Varol, Emilio Ferrara, Clayton A Davis, Filippo Menczer, and Alessandro Flammini. 2017. Online human-bot interactions: Detection, estimation, and characterization. In *Proc. Intl. AAAI Conf. on Web and Social Media (ICWSM)*.
- [44] Onur Varol and Ismail Uluturk. 2018. Deception strategies and threats for online discussions. *First Monday* 23, 5 (2018).
- [45] Onur Varol and Ismail Uluturk. 2020. Journalists on Twitter: self-branding, audiences, and involvement of bots. *Journal of Computational Social Science* 3, 1 (01 Apr 2020), 83–101. <https://doi.org/10.1007/s42001-019-00056-6>
- [46] Nguyen Vo, Kyumin Lee, Cheng Cao, Thanh Tran, and Hongkyu Choi. 2017. Revealing and detecting malicious retweeter groups. In *Proc. of the Intl. Conf. on Advances in Social Networks Analysis and Mining*. 363–368.
- [47] Gang Wang, Manish Mohanlal, Christo Wilson, Xiao Wang, Miriam J. Metzger, Haitao Zheng, and Ben Y. Zhao. 2012. Social Turing Tests: Crowdsourcing Sybil Detection. *CoRR abs/1205.3856* (2012). arXiv:1205.3856
- [48] Kai-Cheng Yang, Onur Varol, Clayton A Davis, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2019. Arming the public with artificial intelligence to counter social bots. *Human Behav. and Emerging Technologies* 1, 1 (2019), 48–61.
- [49] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. 2020. Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1096–1103.